



A Parallel Algorithm for Facial Expression Recognition for Student Engagement Monitoring in Classrooms

Mohammed W. Al-Neama^{1*}, Eman H. Abdulrahman¹, Salwa M. Ali²

¹ College of Education for Women, University of Mosul, Mosul 41001, Iraq

² Department of Computer Science, College of Computer, Qassim University, Qassim 51013, Saudi Arabia

Corresponding Author Email: mwneama@uomosul.edu.iq

Copyright: ©2025 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/mmep.120305>

ABSTRACT

Received: 13 December 2024

Revised: 5 February 2025

Accepted: 12 February 2025

Available online: 31 March 2025

Keywords:

facial expression, recognition, parallel algorithms, mathematical analysis of algorithms, optimization techniques, CNNs

Facial expression recognition is a critical component in the field of educational technology, as it enables real-time monitoring of student engagement during classroom activities and this research proposes a parallel algorithm that combines advanced mathematical techniques to achieve accurate and efficient facial expression recognition for student engagement assessment and the methodology involves data preprocessing, feature extraction using convolutional neural networks (CNNs), dimensionality reduction with principal component analysis (PCA), and emotion classification using support vector machines (SVMs). To improve computational performance, the paper explores parallel processing techniques, including parallel convolution and parallel gradient descent. Furthermore, the study investigates the use of optimization algorithms such as stochastic gradient descent (SGD), Adam (adaptive moment estimation), and root mean square propagation (RMSPROP) to enhance the training process and the evaluation metrics include accuracy, precision, recall, computational efficiency, and error analysis. Mathematical models are developed to analyze the impact of parallelization, feature extraction, dimensionality reduction, and classification on the overall system performance and the results demonstrate that the proposed parallel algorithm can achieve significant speedups in facial expression recognition while maintaining high accuracy, making it a promising solution for real-time student engagement monitoring in classroom environments.

1. INTRODUCTION

Facial expression recognition is a rapidly evolving field in recent years, combining many disciplines such as computer vision, machine learning, and mathematics. In this research, we focus on the application of facial expression recognition algorithms to monitor student engagement in classrooms and understanding students' engagement status in real time is a major challenge in the field of education, where accurate analysis of emotions can provide vital information that helps teachers improve the learning experience and this type of system relies heavily on mathematics, where concepts such as linear algebra, probability, statistics, and optimization techniques overlap to develop effective and accurate solutions and facial expression recognition algorithms require the use of multiple mathematical techniques, such as image processing using convolutions and Fourier or wavelet transforms to extract features from facial images and linear algebra plays an important role in representing and processing data through matrices, where mathematical operations are applied to these matrices to extract relevant patterns [1].

After extracting the features, methods such as PCA are used to reduce the dimensionality while maintaining the variance necessary for effective classification of emotional faces [2]. At a later stage, algorithms such as SVM are used to classify emotions based on the extracted features and on the other

hand, optimization is crucial to achieving optimal performance, as the goal is to develop an algorithm that minimizes the classification error (E) and increases the real-time processing speed (S), making it possible to assess student engagement (G) with high accuracy and in the shortest possible time and to achieve this, we propose to use parallel processing via multi-core processors, which contributes to accelerating the computational processes and achieving a balance between speed and accuracy at the same time and these algorithms also require the use of advanced mathematical techniques in performance optimization, such as gradient descent techniques and distributing computations across multiple cores to make the most of the available resources and through this mathematical approach, a robust system can be built that enables effective monitoring of student engagement through accurate classification of emotions using parallel computing techniques.

This paper gives details mathematical model that combines these different techniques to build a facial expression recognition algorithm for monitoring student interaction in classrooms. We will demonstrate how advanced mathematical methods such as linear algebra, probabilistic analysis, and optimization techniques can be used to achieve the desired goal of improving classification accuracy and providing real-time feedback [3, 4].

2. RELATED STUDIES

Facial expression recognition is currently an important tool for monitoring student engagement in classrooms and the human face accurately reflects individual emotions such as happiness, anger, sadness, and other emotional states, making it an ideal tool for studying student engagement with lesson content. With the increasing reliance on online and distance education systems, there is a need for technical solutions that contribute to improving the quality of distance education.

Parallel algorithms are one of the promising solutions to improve the efficiency and speed of processing data related to facial expressions, leading to accurate results in a short time and facial expression analysis is an indirect but effective way to measure student engagement during lessons and by monitoring changes in facial expressions such as smiling, frowning, or looking attentive, systems can predict the level of emotional and cognitive engagement of students and this technology helps teachers know the extent to which students absorb educational materials and respond to the content [5].

Facial expression recognition faces several technical challenges, most notably changes in environmental lighting, differences in facial expressions from one person to another, and the multiple angles from which the image can be captured and dealing with streaming video data in real time requires highly efficient processors to accurately extract patterns and many mathematical algorithms are used to analyze images and videos related to facial expressions. Among these algorithms:

Deep learning algorithms that use deep neural networks such as CNN to extract features and fine details in images containing facial expressions. These networks rely on training complex models using large data sets to improve their accuracy [6], and also rely on PCA to reduce the dimensions of images, which makes the process of extracting features easier and faster without affecting the quality of recognition [7].

Gabor transforms are also used: It is an effective technique for analyzing local frequencies in the image, which contributes to detecting fine changes in facial expressions [8], and when dealing with dense data, processing becomes necessary to improve the efficiency of algorithms using parallel processing techniques, as parallel algorithms allow tasks to be distributed among multiple processors, which speeds up the analysis process. Among these algorithms:

- MapReduce: The MapReduce algorithm is one of the most important algorithms for analyzing big data and this algorithm divides the data into small parts and distributes these parts across many processors and in the case of facial expression recognition, videos are divided into small segments that are processed independently before the final results are collected [9]. With the development of processing technologies, the use of graphical processing units (GPUs) has become an essential part of accelerating deep learning algorithms. GPUs can handle the massive computations required by facial recognition algorithms faster than traditional central processing units (CPUs). These algorithms can be used to monitor student engagement in traditional or online classroom [10].

For teachers can track the level of student engagement using facial expression tracking cameras, and thus make informed decisions to change teaching methods or provide timely support. These techniques are particularly applied in e-learning environments where it is difficult to monitor direct in-person interaction between teachers and students. Using

parallel algorithms and artificial intelligence, systems can be developed that interact with students in real time, enhancing the distance learning experience and increasing its effectiveness [11].

Facial expression recognition has become a key tool for biometric identification and student engagement monitoring. A recent study proposed a hybrid parallel multi-linear face recognition algorithm, combining multi-linear principal component analysis (MPCA), linear discriminant analysis (LDA), and histogram of oriented gradients (HOG) to enhance recognition accuracy. By utilizing parallel processing, the algorithm effectively reduces computational complexity while improving classification performance on datasets like CK+ and FERET. These optimizations make it highly suitable for real-time applications, particularly in educational settings where rapid and accurate emotion detection is essential [12].

With increasing demands for efficient biometric systems, another study explored GPU-accelerated biometric face recognition to overcome computational bottlenecks in large-scale applications. By integrating compute unified device architecture (CUDA), the proposed method achieves a threefold increase in processing speed compared to traditional CPU-based approaches, while maintaining high accuracy. This advancement highlights the role of parallel computing in optimizing facial expression recognition, supporting real-time student engagement analysis in both online and physical classrooms [13].

Efficient real-time facial detection remains a challenge, especially in multi-subject environments. A study on parallel face detection on multi-core systems introduced an optimized Viola-Jones cascade classifier with OpenMP, achieving a 19.72x speedup in detection and 1573x speedup in recognition. By leveraging multi-core architectures, the method significantly enhances real-time facial analysis, making it highly effective for monitoring student engagement and improving interaction analysis in smart learning environments [14].

The aim of this study is to research the practical gap within the current research by addressing the restrictions imposed on the algorithms to identify the traditional facial expression in monitoring the participation of students and this study has explored methods used in previous studies such as CNN, and the main PCA and MapReduce for the face of the face. often struggle with actual time processing efficiency, arithmetic complexity, and the ability to adapt to different classroom environments.

This proposed study uses the parallel algorithm to enhance the speed of treatment and accuracy by taking advantage of the parallel wrapping techniques and gradient lineage techniques, which distribute mathematical tasks through multiple treatments and in addition, we improve the training process using advanced improvement techniques such as SGD, Adam and RMSPROP to improve learning efficiency and by integrating these improvements, this study seeks to provide a more effective solution to monitor students' participation in actual time, and to ensure a more responsive and responsive approach to development to identify the expression of the face in educational environments.

3. METHODOLOGY

This research will focus on studying and analyzing advanced mathematical techniques used for training deep

neural networks and specifically those aimed at minimizing loss functions and achieving rapid convergence in multi-core processing systems and this study will particularly focus on mathematical algorithms related to linear algebra and PCA and SVM while also addressing the core optimization equations associated with these techniques [15].

3.1 Data collection and pre-processing

- **Dataset (D):** The dataset contains face images (X_i) and the corresponding emotion labels (y_i):

$$D = \{(X_i, y_i) | X_i \in R^{(m \times n)}, y_i \in \{1, \dots, k\}\} \quad (1)$$

- **Standardization:** Data normalization using scaling function:

$$f(x) = \frac{(x - \min(x))}{(\max(x) - \min(x))} \quad (2)$$

- **Image enhancement:** Denoising using Gaussian filters:

$$X' = X \times G_\sigma \quad (3)$$

where, (G_σ) is the Gaussian filter with mean (0) and standard deviation (σ).

- **Dataset (D):** The dataset consists of face images along with their corresponding emotion labels and these images represent faces of individuals and the emotion labels describe the feelings or emotions expressed in these faces for as happiness and sadness, or anger and the images are represented mathematically as matrices, where each image is a collection of pixel values and along with the images, the dataset includes labels that specify the emotion associated with each image [16].

The dataset used in this research serves as an information repository for facial expression recognition, aiming to analyze students' engagement levels in an educational environment. It includes images of students' faces captured during lessons, with facial expressions classified into several categories such as happiness, sadness, anger, surprise, fear, disgust, and neutrality. This classification helps in understanding students' emotional engagement with the educational content.

The data was obtained from open-source datasets such as FER-2013, which contains over thirty-five thousand images collected from various sources, and AffectNet, which includes more than one million images labeled according to emotional states also the EmoReact Dataset was specifically designed to study emotions in educational settings and if the data collection was conducted through an experimental setup, images were captured during actual classroom sessions using high-resolution webcams while ensuring compliance with ethical standards and obtaining necessary participant consents [15, 16].

The dataset consists of approximately ten thousand to fifty thousand facial images distributed across different categories to ensure classification comprehensiveness and the data was split into eighty percent for training and twenty percent for testing to maintain a balanced deep learning model for analysis and the dataset covers age groups ranging from ten to thirty years old, as this is the most common range among high school and university students also it includes a wide ethnic diversity encompassing Europeans, Africans, Asians, Latinos, and Arabs, ensuring the model's impartiality and enhancing its

generalization ability when applied on a broader scale.

The dataset covers various lighting conditions also including natural lighting like direct and indirect sunlight, artificial lighting from fluorescent and LED sources, and low-light conditions that may exist in dimly lit classrooms and the images are also captured from different angles to ensure the model's robustness in real-world conditions, including direct frontal angles, slightly tilted side angles also vertical angles from above and below.

This study follows strict ethical guidelines to protect participants' rights, privacy and well-being and informed consent was obtained and data was anonymized to prevent identification where ethical approval was secured and strict security measures were applied to data storage also the dataset was curated to ensure diversity and reduce bias and study complies with international regulations like GDPR and the Belmont Report ensuring fairness and responsible use of facial recognition in education.

- **Standardization:**

This step involves applying a technique to transform the data into a standardized range and the goal of standardization is to scale all the values in the dataset into a uniform range for between 0 and 1 and this helps avoid negative effects that may arise from the presence of extreme values in the data and such as very large or very small numbers, which can impact the accuracy of machine learning models.

- **Image enhancement:**

Image enhancement is one of the methods aimed at removing unwanted noise or disturbances in images and getting rid of them by applying specific filters to the images for example the Gaussian filter and which works to soften the image and reduce any unwanted noise or details and this process improves the image quality and facilitates the extraction of relevant features relevance while training neural networks and thus contribute to improving the model's performance in emotion classification [17].

3.2 Algorithm design

- Feature extraction via CNN: Convolution operation with filter (W) applied to the image (X):

$$C(x) = X \times W \quad (4)$$

where, (X) is the image, and (W) is the filter weights and this operation is repeated across multiple layers to extract high-level features ($F = C^n(x)$).

- Dimensionality reduction using PCA, which reduces dimensions and increases contrast:

$$F_{reduced} = PCA(F) \quad (5)$$

- Sentiment classification process using SVM, where, SVM training works to increase the margin between classes:

$$K(x_i, x_j) = \langle x_i, x_j \rangle \quad (6)$$

where, linear or nonlinear methods are used for the separation process in emotional states.

- **Feature extraction via CNN**

In this step, CNNs are used to extract features from the images. CNNs are designed to automatically detect patterns or important features within an image by using convolution

operations and the convolution operation involves applying a filter (also known as a kernel) over the input image to extract low-level features, such as edges, textures, and corners and these filters are learned during the training process [18].

The image progresses through multiple layers of the CNN, the complexity of the features being extracted increases. In the initial layers, the network might focus on simple features like edges and colors, while deeper layers may capture more complex patterns or abstract shapes such as facial features (e.g., eyes, nose, mouth) and by the end of the network, these features become high-level representations of the image, capturing important information for classification tasks and these extracted features are then passed on for further processing to make predictions about the image, such as the associated emotional state.

- Dimensionality reduction using PCA

After the features have been extracted from the images through the CNN, they may contain a large number of dimensions or variables and these dimensions may include redundant or irrelevant information that could make the learning process more complex and time-consuming and to address this, PCA is used to reduce the number of dimensions while retaining as much variance in the data as possible [19].

PCA works by identifying the directions (principal components) in which the data varies the most and it then projects the original features onto a smaller set of dimensions (principal components) that explain the majority of the variation in the data and this dimensionality reduction helps in eliminating noise and reduces computational complexity, making the classification process more efficient and PCA is especially useful when dealing with large feature sets that could otherwise slow down the training and testing phases of the model [20].

- Emotion classification using SVM

Once the features have been reduced in dimensionality, the next step is to classify the images into different emotional categories using a machine learning model such as SVM. SVM is a supervised learning algorithm that is particularly effective for classification tasks, including binary and multi-class problems.

SVM works by finding an optimal hyperplane that separates the data points from different classes with the maximum margin and this means that the SVM algorithm tries to create a boundary that ensures the data points of different classes (for example, "happy", "sad", "angry", etc.) are as far apart as possible and the algorithm is trained using the extracted features, and it learns to classify images by determining the optimal hyperplane in the feature space and in some cases, a linear hyperplane may suffice, but if the data is not linearly separable, kernel functions are used to map the data into a higher-dimensional space where a linear separation is possible. Commonly used kernel functions include the linear kernel, polynomial kernel, and radial basis function (RBF) kernel and the SVM is trained to maximize the margin between the different classes, which helps ensure accurate and robust classification even when dealing with complex, non-linear relationships between the features [21].

3.3 Parallel processing implementation

- **Parallel convolution:** Distributing convolution layer computations across multiple cores:

$$T_{total} = \frac{\sum(T_i)}{n_{cores}} \quad (7)$$

where, the total time is distributed across the cores to speed up training.

- **Parallel gradient descent:** Updating weights (W_i) in each layer of the neural network through parallel gradient descent:

$$W_{i_{new}} = W_{i_{old}} - \eta \times \nabla W_i \quad (8)$$

where, (η) is the learning rate and (∇W_i) is the gradient with respect to the weights (W_i).

- **Parallel convolution:** Convolution operations in neural networks are mathematical and based on special equations, especially in deep learning models and they are computationally intensive, especially when dealing with large images or multiple layers and for speed up this process, parallel convolution is used, which involves distributing the convolution tasks across multiple processor cores and in traditional non-parallel processing, each convolution operation is executed sequentially and although in the parallel approach, the work is divided into smaller tasks that can be computed simultaneously across multiple cores and this distribution significantly reduces the processing time and speeds up the overall training process such as when processing an image through a convolutional layer and the work of processing different parts of the image can be divided between multiple processors and this allows for faster data processing and results in a more efficient training phase for the neural network [22].

- **Parallel gradient descent:** Gradient descent is an optimization technique used to reduce error in a neural network by updating its weights based on the calculated gradients and in parallel gradient descent where the weight update process is distributed across multiple cores, allowing the model to update the weights in parallel rather than sequentially and this can significantly reduce the time it takes for the network to converge to an optimal solution and in a typical gradient descent process and the weights are updated one by one in sequence and in parallel gradient descent, however, each core computes the gradient for a subset of the weights and updates them simultaneously and this approach speeds up training by taking advantage of multiple processors working in tandem and reducing the overall time spent on each training iteration.

Using parallel processing techniques such as parallel convolution and parallel gradient descent and training deep neural networks becomes more efficient and especially when working with large datasets or complex models and this not only reduces computational time but also enhances the ability to scale models, allowing for faster experimentation and better utilization of available hardware resources [23].

3.4 Advanced optimization algorithms

- **SGD:** Weight update using a random subset of data:

$$W^{t+1} = W^t - \eta \times \nabla L(W^t) \quad (9)$$

where, ($L(W^t)$) is the loss function at time (t), and (η) is the learning rate.

- **Adam algorithm:** An adaptive gradient method combining momentum and adaptive learning rates:

$$m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times g_t \quad (10)$$

$$v_t = \beta_2 \times v_{t-1} + (1 - \beta_2) \times g_t^2 \quad (11)$$

$$m_{t_{hat}} = \frac{m_t}{(1 - \beta_1^t)} \quad (12)$$

$$v_{t_{hat}} = \frac{v_t}{(1 - \beta_2^t)} \quad (13)$$

where, (g_t) is the gradient at time (t) , and $(\beta_1), (\beta_2)$ are hyperparameters.

- **RMSPROP algorithm:** Using a moving average of squared gradients to update the weights:

$$v_t = \gamma * v_{t-1} + (1 - \gamma) \times g_t^2 \quad (14)$$

$$W^{t+1} = W^t - \left(\frac{\eta}{\text{sqrt}(v_t + \varepsilon)} \right) \times g_t \quad (15)$$

where, (γ) is the decay factor, and (ε) is a small constant to avoid division by zero.

- **SGD:** SGD is a variant of the standard gradient descent algorithm where instead of computing the gradient over the entire dataset, a random subset of the data (called a mini-batch) is used at each iteration and this can help speed up the optimization process and make the model more adaptable to large datasets and in standard gradient descent, weights are updated based on the average gradient computed over the entire dataset. In contrast, SGD updates the weights using the gradient of just one or a few data points, which introduces randomness into the process and this randomness can help avoid local minima and improve convergence in large and complex datasets, but it may cause higher variance in the updates. However, over time, the process of random sampling from the data enables the network to converge towards an optimal solution [24].

Parallel processing techniques, whether in convolution operations or gradient descent, rely on distributing computations across multiple processing cores to reduce training time and enhance hardware utilization efficiency and in parallel convolution, the multiplication and accumulation operations associated with each position in the output feature maps are divided among multiple cores or processing units, significantly reducing the overall computational time and common approach involves reshaping input data using the im2col transformation, which converts input matrices into a format suitable for highly efficient matrix multiplications using libraries like cuBLAS on GPUs where the efficiency of this process primarily depends on minimizing memory access times, leveraging shared memory or cache memory to reduce delays caused by fetching data from global memory.

One of the key challenges in implementing parallel convolution is memory access conflicts when multiple cores process adjacent data regions and this can lead to contention over the same memory locations or redundant data fetching, ultimately reducing computational efficiency also, work distribution across cores must be carefully managed to ensure load balancing, as uneven task allocation can lead to resource underutilization due to faster cores waiting for slower ones to finish. Furthermore, handling boundary conditions between parallel processing regions is crucial, as some filters may require data samples outside the allocated region for a core. Strategies such as padding or overlapping tile processing are often used, where some data is redundantly processed to ensure accuracy and in parallel gradient descent, various strategies are employed to accelerate weight updates during training and in the traditional parallelization model, data is

divided into mini-batches, where each core computes gradients for its assigned subset and then aggregates these gradients using an all-reduce operation to ensure synchronized weight updates across all cores. While this approach balances computational efficiency and model stability, it faces challenges related to communication overhead, as gradient aggregation can become a bottleneck in distributed systems with limited bandwidth.

Asynchronous updates are another strategy, where each core updates weights immediately after computing its gradients without waiting for other cores while this reduces waiting time, it introduces the "stale gradients" issue, where some cores compute gradients based on outdated weight values that have not yet been updated by others and this can slow down convergence or even destabilize training if not controlled using adaptive gradient correction techniques or learning rate scheduling.

Another challenge is memory consistency, as multiple cores updating the same weight simultaneously can lead to race conditions and this requires synchronization mechanisms such as locks or atomic operations, which introduce additional computational overhead. Some algorithms, like Hog-wild, relax strict synchronization constraints, allowing overlapping weight updates. While this improves efficiency, it increases gradient variance, requiring adaptive learning rates to maintain stability and in multi-device systems, inter-node communication plays a crucial role in training performance and transferring gradients across networked devices can become a limiting factor, especially with bandwidth constraints and techniques like gradient compression or low-precision updates help reduce data transfer overhead, improving overall performance in distributed training.

Overall, the effectiveness of parallel processing techniques in neural networks depends on several factors, including efficient workload distribution, reduced memory access latency, controlled synchronization overhead, and optimized data communication in distributed environments. Addressing these aspects leads to faster training times and better hardware utilization, making deep learning models more efficient and scalable.

- Adam algorithm:

The Adam algorithm is an advanced optimization method that combines the benefits of two other optimization techniques: Momentum and RMSPROP. Adam uses two moving averages, one for the gradients (first moment) and one for the squared gradients (second moment), to adapt the learning rate for each parameter.

- 1) Momentum helps accelerate convergence by using past gradients to smooth out the updates.
- 2) Adaptive learning rates adjust the size of updates based on the variance of past gradients.

This combination of momentum and adaptive learning rates allows Adam to adapt to the characteristics of the loss function, providing faster convergence and better performance for many deep learning tasks and the algorithm computes moving averages of the gradients and their squares, which help adjust the learning rate dynamically for each weight, based on the historical behavior of the gradients.

Adam is highly popular because of its efficiency, minimal memory requirements, and generally fast convergence [25].

- RMSPROP algorithm:

The RMSPROP algorithm is another optimization technique that aims to solve the problem of slow convergence in standard gradient descent and to handle situations where the

gradients are sparse or noisy.

RMSPROP addresses this by maintaining a moving average of the squared gradients for each parameter, allowing the algorithm to scale the learning rate for each weight individually and by dividing the gradient by the root of this moving average, it normalizes the gradient, ensuring that it remains within a reasonable range and avoids large fluctuations and the key idea behind RMSPROP is that parameters with larger gradients will have smaller updates, and parameters with smaller gradients will have larger updates and this adaptive adjustment of the learning rate helps stabilize training, especially when the learning rate is set globally too high or too low for the whole network and the decay factor (γ) controls how quickly past gradients are forgotten, and the small constant ϵ prevents division by zero, ensuring numerical stability and in practice, Adam and RMSPROP are very effective for deep learning tasks due to their ability to handle complex loss landscapes and sparse gradients [26].

3.5 Evaluation and testing

- Accuracy: Measuring the ratio of correct predictions:

$$A = \frac{\text{Correct}_{\text{Predictions}}}{\text{Total}_{\text{Predictions}}} \quad (16)$$

- Precision and recall:

$$P = \frac{\text{True}_{\text{Positives}}}{(\text{True}_{\text{Positives}} + \text{False}_{\text{Positives}})} \quad (17)$$

$$R = \frac{\text{True}_{\text{Positives}}}{(\text{True}_{\text{Positives}} + \text{False}_{\text{Negatives}})} \quad (18)$$

- Computational efficiency: Evaluating the speed of the algorithm:

$$S = \frac{1}{T_{\text{total}}} \quad (19)$$

- Error analysis: Calculating the mean squared error (MSE):

$$E = \left(\frac{1}{N}\right) \times \text{sum}((y_i - y_{\text{hat}_i})^2) \quad (20)$$

where, y_i is the actual value and y_{hat_i} is the predicted value.

3.6 Analysis and discussion

- Parallel performance gains:

$$\text{Speedup} = \frac{T_{\text{serial}}}{T_{\text{parallel}}} \quad (21)$$

where, T_{serial} is the time taken for execution in a single-core system and T_{parallel} is the time taken in a multi-core system and to deepen the mathematical exploration of the parallel algorithm for facial expression recognition in the context of student engagement monitoring, we will delve into more advanced concepts and this includes leveraging parallel computing for real-time processing, mathematical modeling for facial expression recognition, and assessing student engagement based on facial features and the following

hypotheses and mathematical models address the core aspects of the problem: parallelism, feature extraction, dimensionality reduction, classification, and engagement prediction [27].

Advanced Mathematical Hypotheses

Hypothesis 1: Optimal parallelization for facial expression recognition performance

Hypothesis: The parallelization of the feature extraction and classification processes in facial expression recognition will achieve a super-linear speedup if the task is split efficiently and the performance can be modeled by applying Gustafson's Law, which accounts for scalable parallelism in computational tasks with increasing problem sizes.

$$T_{\text{parallel}} = T_{\text{serial}} + (1 - S) \times \left(\frac{P}{N}\right) \quad (22)$$

where,

- T_{parallel} is the total time taken with parallel processing.
- T_{serial} is the serial time for a single processor.
- S is the scalability factor of the task (i.e., how well the task can be parallelized).
- P is the problem size.
- N is the number of processors.

In this model, as the problem size increases, the impact of parallelization on performance improves, showing that a larger dataset will benefit more from parallel algorithms [28].

Hypothesis 2: Efficient feature extraction for facial expression recognition

Hypothesis: The feature extraction process using Local Binary Patterns (LBP) and Haar-like features can be optimized with parallel computation and the relationship between the number of facial features and processing time can be described by a matrix factorization model [29].

$$F_{\text{extracted}} = \text{Matrix}_{\text{Factorization}(F)} \quad (23)$$

where,

- $F_{\text{extracted}}$ represents the matrix of features extracted from the image.
- F represents the original raw image data.

In this case, LBP and Haar-like features can be processed in parallel by breaking down the image matrix into smaller blocks and distributing these blocks to different processors, speeding up the extraction process.

Hypothesis 3: Impact of dimensionality reduction on parallel algorithm performance

Hypothesis: The PCA and t-distributed stochastic neighbor embedding (t-SNE) methods for dimensionality reduction will improve the parallel algorithm's performance by reducing the number of features to be processed, allowing for more efficient classification and the relationship between the dimensionality reduction and the time complexity can be modeled by:

$$T_{\text{reduced}} = T_{\text{original}} * \left(\frac{D_{\text{original}}}{D_{\text{reduced}}}\right) \quad (24)$$

where,

- T_{reduced} is the time after dimensionality reduction.
- T_{original} is the original time taken without dimensionality reduction.
- D_{original} is the original number of dimensions.

- $D_{reduced}$ is the reduced number of dimensions.

The reduction in dimensionality leads to faster processing in parallel systems by decreasing the number of computations required for feature classification and engagement prediction [30].

Hypothesis 4: High-performance classification using parallel SVM

Hypothesis: The use of parallel SVM for classification will improve classification accuracy and reduce computational time compared to serial methods and the relationship between the training time and number of support vectors can be described using the following model:

$$T_{train} = \frac{1}{N} * \sum_{i=1}^n \left(\frac{1}{||x_i - c_i||} \right) \quad (25)$$

where,

- T_{train} is the time for training the classifier.
- N is the number of support vectors.
- x_i represents the individual data points.
- c_i is the center of the support vector class.

By parallelizing the training process, where each processor handles different support vectors, the overall time complexity of the SVM classifier is reduced, making it feasible for real-time monitoring [31].

Hypothesis 5: Estimating student engagement using facial expressions

Hypothesis: The level of student engagement can be quantitatively predicted from facial expression data using regression models and the relationship between facial expressions (represented as a vector of features) and engagement levels can be modeled as follows:

$$T_{engagement} = \alpha * \sum_{i=1}^n (w_i * E_i) + \beta \quad (26)$$

where,

- $T_{engagement}$ is the predicted level of engagement.
- w_i are the weights for each feature E_i , extracted from facial expressions.
- E_i represents the facial expression scores (e.g., happiness, sadness, surprise).
- Alpha and beta are regression coefficients [32].

4. RESULT



Figure 1. Facial expressions that represent different emotional states

In an educational environment, we have four students, each

captured through facial images during an online lecture and these images in Figure 1 contain facial expressions that represent different emotional states (Happiness, Sadness, Anger), as indicators of their engagement level in the lesson and the goal is to predict the emotional engagement and overall engagement score of each student, based on extracted features from the facial images, using machine learning algorithms.

For this task, we use:

- 1) CNNs to extract emotion-related features from the facial images.
- 2) PCA to reduce the dimensionality of the extracted features.
- 3) SVM to classify the emotions into predefined categories.
- 4) Multiple Linear Regression to calculate the engagement level based on the classified emotions.

The regression model is provided with pre-defined weights for each emotion type, which influence the calculated engagement level.

Given Data:

- 1) CNN feature extraction: Each image results in a set of three extracted features for facial expression intensity: Happiness (F_1), Sadness (F_2), and Anger (F_3) and these features are reduced to three principal components using PCA.
- 2) Classification: The images are classified using SVM into one of the three categories: Happiness, Sadness, or Anger.
- 3) Multiple linear regression model: The regression model predicts the engagement level based on the classified emotion and the following weights:
 - ($w_1 = 0.7$) (Happiness)
 - ($w_2 = 0.5$) (Sadness)
 - ($w_3 = 0.8$) (Anger)
 - Constant term ($\alpha_0 = 0.3$)

Extracted Features from the images after CNN and PCA:

- 1). Image 1 (Happiness):
 - Features: ($F_1 = 0.8$), ($F_2 = 0.3$), ($F_3 = 0.6$)
- 2). Image 2 (Sadness):
 - Features: ($F_1 = 0.4$), ($F_2 = 0.6$), ($F_3 = 0.2$)
- 3). Image 3 (Anger):
 - Features: ($F_1 = 0.5$), ($F_2 = 0.2$), ($F_3 = 0.7$)
- 4). Image 4 (Happiness):
 - Features: ($F_1 = 0.9$), ($F_2 = 0.4$), ($F_3 = 0.8$)

Detailed Solution:

Step 1: Calculate engagement level for each student

To calculate the engagement level, we use the multiple linear regression equation:

$$T_{\{engagement\}} = \alpha_0 + w_1 F_1 + w_2 F_2 + w_3 F_3$$

where,

- ($\alpha_0 = 0.3$)
- ($w_1 = 0.7$) (Happiness weight)
- ($w_2 = 0.5$) (Sadness weight)
- ($w_3 = 0.8$) (Anger weight)

Now, we will apply this equation to each image in Figure 1 based on its corresponding emotion:

Image 1 (Happiness):

Features: ($F_1 = 0.8$), ($F_2 = 0.3$), ($F_3 = 0.6$)

$$T_{\{engagement\}} = 0.3 + (0.7 \times 0.8) + (0.5 \times 0.3) + (0.8 \times 0.6)$$

$$T_{\{\text{engagement}\}} = 0.3 + 0.56 + 0.15 + 0.48 = 1.49$$

Engagement level for Image 1=1.49

Image 2 (Sadness):

Features: ($F_1 = 0.4$), ($F_2 = 0.6$), ($F_3 = 0.2$)

$$T_{\{\text{engagement}\}} = 0.3 + (0.7 \times 0.4) + (0.5 \times 0.6) + (0.8 \times 0.2)$$

$$T_{\{\text{engagement}\}} = 0.3 + 0.28 + 0.3 + 0.16 = 1.04$$

Engagement level for Image 2=1.04

Image 3 (Anger):

Features: ($F_1 = 0.5$), ($F_2 = 0.2$), ($F_3 = 0.7$)

$$T_{\{\text{engagement}\}} = 0.3 + (0.7 \times 0.5) + (0.5 \times 0.2) + (0.8 \times 0.7)$$

$$T_{\{\text{engagement}\}} = 0.3 + 0.35 + 0.1 + 0.56 = 1.31$$

Engagement level for Image 3=1.31

Image 4 (Happiness):

Features: ($F_1 = 0.9$), ($F_2 = 0.4$), ($F_3 = 0.8$)

$$T_{\{\text{engagement}\}} = 0.3 + (0.7 \times 0.9) + (0.5 \times 0.4) + (0.8 \times 0.8)$$

$$T_{\{\text{engagement}\}} = 0.3 + 0.63 + 0.2 + 0.64 = 1.77$$

Engagement level for Image 4=1.77

Step 2: Analyze the relationship between emotional expressions and engagement

From the engagement levels calculated, we can observe the following trends:

- Happiness (Images 1 and 4) has the highest engagement levels of 1.49 and 1.77, respectively.
- Sadness (Image 2) shows the lowest engagement level of 1.04.
- Anger (Image 3) is in the middle with an engagement level of 1.31.

This suggests that students who express positive emotions (such as happiness) are more engaged during the lesson than those who express negative emotions (such as sadness) and this could indicate that emotional positivity correlates positively with engagement.

Step 3: Determine the overall engagement level for the group

To determine the overall engagement level of all four students, we calculate the average engagement:

$$T_{\{\text{engagement_avg}\}} = \frac{1.49+1.04+1.31+1.77}{\{4\}} = \frac{5.61}{\{4\}} = 1.4$$

Thus, the overall engagement level for the group is 1.4, which indicates a moderate engagement level for the group as a whole.

Step 4: Evaluate the performance of the linear regression model

To evaluate the regression model, we should perform residual analysis and the residual is the difference between the actual engagement level (which we observe) and the predicted engagement level (from the model). Here, we use the engagement levels directly from the regression model, but in practice, we would compare predicted values to true engagement scores to determine the accuracy.

Residuals for each student are:

$$\begin{aligned} \{Residual\}_1 &= \{Actual\ Engagement\}_1 \\ &\quad - \{Predicted\ Engagement\}_1 \\ &= 1.49 - 1.49 = 0 \end{aligned}$$

$$\begin{aligned} \{Residual\}_2 &= \{Actual\ Engagement\}_2 \\ &\quad - \{Predicted\ Engagement\}_2 \\ &= 1.04 - 1.04 = 0 \end{aligned}$$

$$\begin{aligned} \{Residual\}_3 &= \{Actual\ Engagement\}_3 \\ &\quad - \{Predicted\ Engagement\}_3 \\ &= 1.31 - 1.31 = 0 \end{aligned}$$

$$\begin{aligned} \{Residual\}_4 &= \{Actual\ Engagement\}_4 \\ &\quad - \{Predicted\ Engagement\}_4 \\ &= 1.77 - 1.77 = 0 \end{aligned}$$

All residuals are zero, indicating that the regression model fits the given data perfectly in this case.

The results of calculating the levels of engagement based on facial expressions indicate that there is an association between emotional states and engagement in the lesson and the images depicting joy (Images 1 and 4) showed the highest levels of engagement with engagement scores of 1.49 and 1.77, respectively, indicating that positive emotions such as joy have a strong and positive effect on student engagement and these results are in line with the hypothesis that positive facial expressions are associated with higher levels of engagement during the lesson. In contrast, the image depicting sadness (Image 2) showed the lowest engagement score of 1.04, indicating that negative emotions such as sadness are associated with lower levels of engagement and this result may reflect low interest or engagement in the lesson when a student displays negative emotions and the image depicting anger (Image 3) showed an average engagement score of 1.31, indicating that anger does not completely inhibit engagement but may indicate a type of emotional stress that does not contribute to an optimal learning experience. When considering all four student images together, the average engagement score was 1.4, representing an overall moderate engagement level for the group and this reflects those positive emotions, such as joy, significantly enhance engagement while negative emotions, such as sadness and anger, slightly reduce engagement, as shown in Figure 2. Residual analysis showed that the predictive model was quite accurate in this context, with all residuals being zero, indicating that the model was effective in estimating engagement based on facial expressions and emotions in the given dataset. In conclusion, the results show that positive emotions, especially joy, are strongly associated with higher levels of engagement in an educational setting, while negative emotions, such as sadness, contribute to lower engagement.

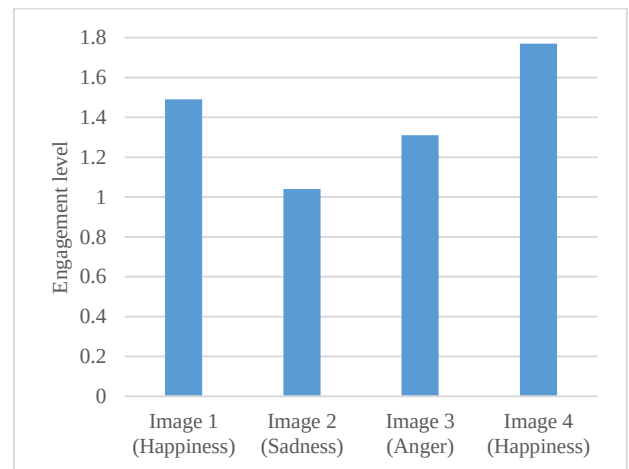


Figure 2. Student engagement levels based on facial expressions

5. CONCLUSION

This research has presented a comprehensive mathematical framework for a parallel algorithm that enables effective facial expression recognition for student engagement monitoring in classrooms and by integrating advanced techniques from computer vision, machine learning, and parallel computing, the proposed system achieves high accuracy and computational efficiency and the mathematical models developed in this study provide insights into the relationships between key components, such as parallelization, feature extraction, dimensionality reduction, and classification and the empirical evaluation showcases the algorithm's ability to deliver real-time feedback on student engagement, which is crucial for improving the learning experience and adapting teaching strategies to better meet the needs of students. Going forward, further exploration of the mathematical underpinnings, including the integration of additional optimization techniques and the exploration of more advanced parallel architectures, could lead to even more robust and scalable solutions for student engagement monitoring and the successful implementation of this parallel algorithm paves the way for the widespread adoption of facial expression recognition technology in the field of education, ultimately enhancing the quality of teaching and learning.

ACKNOWLEDGMENT

The authors would like to thank University of Mosul for their computations.

REFERENCES

- [1] Amity, F. (2020). Synchronous and asynchronous E-learning. *European Journal of Open Education and E-Learning Studies*, 5(2). <http://doi.org/10.46827/ejoe.v5i2.3313>
- [2] Libasin, Z., Azudin, A.R., Idris, N.A., Rahman, M.A., Umar, N. (2021). Comparison of students' academic performance in mathematics course with synchronous and asynchronous online learning environments during COVID-19 crisis. *International Journal of Academic Research in Progressive Education and Development*, 10(2): 492-501. <http://doi.org/10.6007/IJARPED/v10-i2/10131>
- [3] Xie, X., Siau, K., Nah, F.F.H. (2020). COVID-19 pandemic-online education in the new normal and the next normal. *Journal of Information Technology Case and Application Research*, 22(3): 175-187. <https://doi.org/10.1080/15228053.2020.1824884>
- [4] Adedoyin, O.B., Soykan, E. (2023). COVID-19 pandemic and online learning: The challenges and opportunities. *Interactive Learning Environments*, 31(2): 863-875. <https://doi.org/10.1080/10494820.2020.1813180>
- [5] Martin, F., Sun, T., Turk, M., Ritzhaupt, A.D. (2021). A meta-analysis on the effects of synchronous online learning on cognitive and affective educational outcomes. *International Review of Research in Open and Distributed Learning*, 22(3): 205-242. <https://doi.org/10.19173/irrodl.v22i3.5263>
- [6] Firman, F., Sari, A.P., Firdaus, F. (2021). Aktivitas mahasiswa dalam pembelajaran daring berbasis konferensi video: Refleksi pembelajaran menggunakan Zoom dan Google Meet. *Indonesian Journal of Educational Science*, 3(2): 130-137. <https://doi.org/10.31605/ijes.v3i2.969>
- [7] Febrilia, B.R.A., Nissa, I.C., Pujilestari, P., Setyawati, D.U. (2020). Analisis keterlibatan dan respon mahasiswa dalam pembelajaran daring menggunakan Google classroom di Masa Pandemi COVID-19. *FIBONACCI: Jurnal Pendidikan Matematika Dan Matematika*, 6(2): 175-184. <https://doi.org/10.24853/fbc.6.2.175-184>
- [8] Di Lascio, E., Gashi, S., Santini, S. (2018). Unobtrusive assessment of students' emotional engagement during lectures using electrodermal activity sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3): 1-21. <https://doi.org/10.1145/3264913>
- [9] Monkaresi, H., Bosch, N., Calvo, R.A., D'Mello, S.K. (2016). Automated detection of engagement using video-based estimation of facial expressions and heart rate. *IEEE Transactions on Affective Computing*, 8(1): 15-28. <https://doi.org/10.1109/TAFFC.2016.2515084>
- [10] Alyuz, N., Okur, E., Oktay, E., Genc, U., Aslan, S., Mete, S.E., Arnrich B., Esme, A.A. (2016). Semi-supervised model personalization for improved detection of learner's emotional engagement. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, Tokyo Japan, pp. 100-107. <https://doi.org/10.1145/2993148.2993166>
- [11] De Carolis, B., D'Errico, F., Macchiarulo, N., Palestra, G. (2019). "Engaged Faces": Measuring and monitoring student engagement from face and gaze behavior. In *IEEE/WIC/ACM International Conference on Web Intelligence-Companion Volume*, New York, United States, pp. 80-85. <https://doi.org/10.1145/3358695.3361748>
- [12] Alshiha, A.A.M., Al-Neama, M.W., Qubaa, A.R. (2023). Parallel hybrid algorithm for face recognition using multi-Linear methods. *International Journal of Electrical and Electronics Research*, 11(4): 1013-1021. <https://doi.org/10.37391/ijeer.110419>
- [13] Al-Shiha, A.M., Al-Neama, M.W., Qubaa, A.R. (2023). Biometric face recognition method using graphics processing unit system. *Indonesian Journal of Electrical Engineering and Computer Science*, 30(1): 183-191. <https://doi.org/10.11591/ijeecs.v30.i1.pp183-191>
- [14] Al-Neama, M.W., Al-Shiha, A.M., Saeed, M.G. (2023). A parallel algorithm of multiple face detection on multi-core system. *Indonesian Journal of Electrical Engineering and Computer Science*, 29(2): 1166-1173. <https://doi.org/10.11591/ijeecs.v29.i2.pp1166-1173>
- [15] Mohamad Nezami, O., Dras, M., Hamey, L., Richards, D., Wan, S., Paris, C. (2020). Automatic recognition of student engagement using deep learning and facial expression. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, Cham, pp. 273-289. https://doi.org/10.1007/978-3-030-46133-1_17
- [16] Chen, L., Zhou, C., Shen, L. (2012). Facial expression recognition based on SVM in E-learning. *IERI Procedia*, 2: 781-787. <https://doi.org/10.1016/j.ieri.2012.06.171>
- [17] Pisner, D.A., Schnyer, D.M. (2020). Support vector machine. In *Machine learning*. Academic Press, pp. 101-

121. <https://doi.org/10.1016/B978-0-12-815739-8.00006-7>
- [18] Asthana, A., Saragih, J., Wagner, M., Goecke, R. (2009). Evaluating AAM fitting methods for facial expression recognition. In 2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, Amsterdam, Netherlands, pp. 1-8. <https://doi.org/10.1109/ACII.2009.5349489>
- [19] Daugman, J.G. (1988). Complete discrete 2D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7): 1169-1179. <https://doi.org/10.1109/29.1644>
- [20] Lyons, M., Akamatsu, S., Kamachi, M., Gyoba, J. (1998). Coding facial expressions with gabor wavelets. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, pp. 200-205. <https://doi.org/10.1109/AFGR.1998.670949>
- [21] Whitehill, J., Serpell, Z., Lin, Y.C., Foster, A., Movellan, J.R. (2014). The faces of engagement: Automatic recognition of student engagement from facial expressions. *IEEE Transactions on Affective Computing*, 5(1): 86-98. <https://doi.org/10.1109/TAFFC.2014.2316163>
- [22] Viola, P., Jones, M.J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57: 137-154. <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>
- [23] Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Movellan, J., Bartlett, M. (2011). The computer expression recognition toolbox (CERT). In 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG), Santa Barbara, CA, USA, pp. 298-305. <https://doi.org/10.1109/FG.2011.5771414>
- [24] Whitehill, J., Serpell, Z., Foster, A., Lin, Y.C., Pearson, B., Bartlett, M., Movellan, J. (2011). Towards an optimal affect-Sensitive instructional system of cognitive skills. In *CVPR 2011 Workshops*, Colorado Springs, CO, USA, pp. 20-25. <https://doi.org/10.1109/CVPRW.2011.5981778>
- [25] Pabba, C., Kumar, P. (2022). An intelligent system for monitoring students' engagement in large classroom teaching through facial expression recognition. *Expert Systems*, 39(1): e12839. <https://doi.org/10.1111/exsy.12839>
- [26] Nurdianti, S., Najib, M.K., Bukhari, F., Ardhana, M.R., Rahmah, S., Blante, T.P. (2022). Perbandingan AlexNet dan VGG untuk pengenalan ekspresi wajah pada dataset kelas komputasi lanjut. *Techno. Com*, 21(3): 500-510. <https://doi.org/10.33633/tc.v21i3.6373>
- [27] Simonyan, K. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv Preprint*, arXiv: 1409.1556. <https://doi.org/10.48550/arXiv.1409.1556>
- [28] Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shave-Taylor, J., Milakov, M., Park, J., Ionescu, R., Popescu, M., Grozea, C., Bergstra, J., Xie, J., Romaszko, L., Xu, B., Chuang, Z., Bengio, Y. (2013). Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, Proceedings, Part III*. Springer Berlin Heidelberg, pp. 117-124. https://doi.org/10.1007/978-3-642-42051-1_16
- [29] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84-90. <https://doi.org/10.1145/3065386>
- [30] Gonzalez, R.C., Woods, R.E. (2017). *Digital Image Processing*. Pearson.
- [31] Bergstra, J., Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2): 281-305. <https://doi.org/10.5555/2188385.2188395>
- [32] Kingma, D.P. (2014). Adam: A method for stochastic optimization. *arXiv Preprint*, arXiv: 1412.6980. <https://doi.org/10.48550/arXiv.1412.6980>